第 37 卷第 1 期
录用定稿
计算机应用研究
Application Research of Computers
Vol. 37 No. 1
Accepted Paper

基于比特可分性的 PRIDE 和 RoadRunneR 积分区分器搜索 *

李艳俊，赵京鸣

(北京电子科技学院 信息安全系，北京 100070)

摘 要：PRIDE 和 RoadRunneR 是近几年提出的两种轻量级分组密码算法，在 2016 亚密会上，向泽军等提出利用基于比特可分性的 MILP(混合整数线性规划)模型搜索积分区分器。利用该思想，针对两种不同类型的轻量级分组密码算法，为了评估该算法积分性质，验证新方法的实用性，根据其不同密码算法结构分别建立 MILP 模型，利用 Gurobi 优化器求解此模型，搜索可用的积分区分器。结果分别得到 9 轮和 5 轮的积分区分器，是 PRIDE 和 RoadRunneR 目前已知最长的积分区分器，利用该区分器可进行更多轮的积分攻击。

关键词：PRIDE；RoadRunneR；比特可分性；MILP 模型；积分区分器

中图分类号：TP309.2　　doi: 10.19734/j.issn.1001-3695.2018.06.0473

Integral distinguisher search of PRIDE and RoadRunneR based on bit-based division property

Li Yanjun, Zhao Jingming

(*Dept. of Information Security*, *Beijing Electronic Science & Technology Institute*, *Beijing* 100070, *China*)

Abstract: PRIDE and RoadRunneR are two lightweight block ciphers proposed in recent years. At 2016 ASIACRYPT, Xiang Zejun proposed using MILP (mixed integer linear programming) model based on bit-based division to search integral distinguisher. This paper applies this idea to two lightweight block cipher that two different types of algorithms. In order to evaluate their integral properties, MILP models are built according to their different structures, and the useful integral distinguisher can be searched by using Gurobi optimizer to solve this model. Results, 9 rounds and 5 rounds of integral distinguisher are obtained respectively, which is the longest integral distinguisher of PRIDE and RoadRunneR . More rounds of integral attack can be made by using the distinguisher.

Key words: PRIDE; RoadRunneR; Bit-based Division; MILP model; Integral distinguisher

1　相关工作

1.1　PRIDE 和 RoadRunneR 算法简介

PRIDE 算法是 Albrecht 等人在 2014 美密会上提出的分组长度为 64 bit，密钥长度为 128 bit 的 SPN 结构轻量级分组密码算法，共迭代 20 轮。64 bit 的轮函数输入被分为 16 个半字节(Nibble)，与轮密钥做异或运算，然后分别并行进入 S 层，即并置 16 个相同的 S 盒（见表 1），最后经过线性层，具体的轮函数如图 1 所示。

表 1　PRIDE 算法 S 盒

Table 1　S box of PRIDE algorithm

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S[x]$	0	4	8	f	1	5	e	9	2	7	a	c	b	d	6	3

PRIDE 的线性层 L 可以被分成 3 个子层，比特置换层 P，矩阵层 M，逆比特置换层 P^{-1} .$\boldsymbol{M}$ 为一个 64×64 的矩阵，由四个 16×16 的矩阵 L_0,L_1,L_2,L_3 构成；P 为 64 位的比特置换操作。其中，L_0,L_1,L_2,L_3 的具体矩阵以及 PRIDE 的其他信息详见文献[1]。

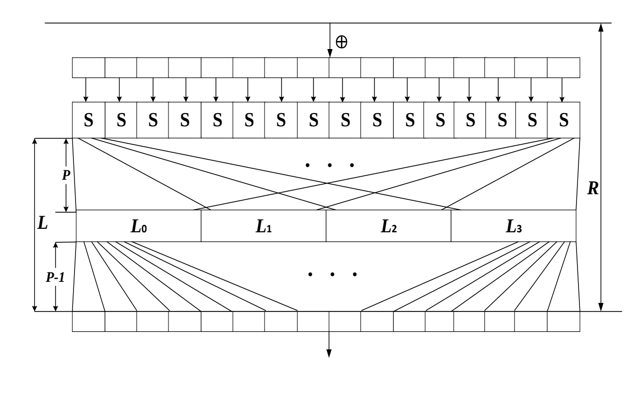


图 1　PRIDE 算法示意图

Fig.1　Schematic diagram of PRIDE algorithm

轻量级分组密码算法 RoadRunneR 在 2015 年提出。总体采用了 Feistel 结构，轮函数采用 SPN 结构。分组长度 64 比特，密钥长度为 80/128 bit，分别迭代 10/12 轮。轮函数 F 由 4 轮的 SPN 函数构成，即 3 个 SLK 函数加 1 个 S 层，S 盒见表 0-2。

收稿日期：2018-06-06；**修回日期：**2018-07-18　**基金项目：**中央高校基本科研业务费资助项目（2017LG04）

作者简介：李艳俊（1979-），女，山西晋城人，副教授，博士，主要研究方向为信息安全、密码函数（lyj@besti.edu.cn）；赵京鸣（1993-），男，甘肃定西人，硕士研究生，主要研究方向为分组密码分析.

S、L 和 K 分别代表 S 层，L 层和轮密钥加层。具体的算法结构和轮函数见图 2，图中左边为算法整体 Feistel 结构，右上为 SPN 型轮函数 F，右下为 SLK 函数。

表 2　RoadRunneR 算法 S 盒

Table 2　S box of RoadRunneR algorithm

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S[x]$	0	8	6	d	5	f	7	c	4	e	2	3	9	1	b	a

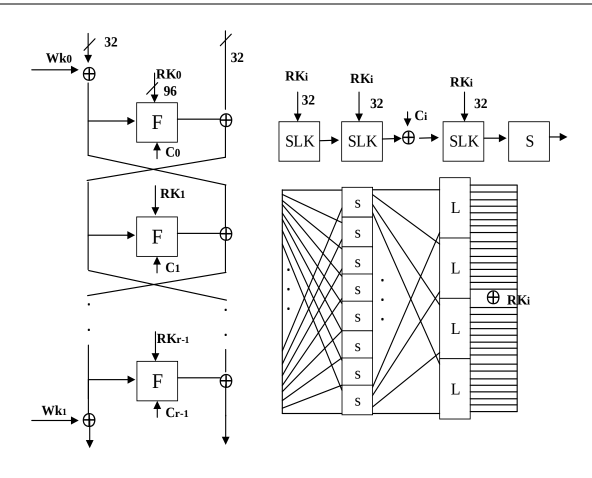


图 2　RoadRunner 算法示意图

Fig.2　Schematic diagram of RoadRunner algorithm

RoadRunneR 线性变换层可表示为 $L(x)=(x)\oplus(x<<<1)\oplus(x<<<2)$，其中 $x<<<1$ 表示字节 x 向左循环移位 1 比特。 $L(x)$ 用 8×8 矩阵可表示为

$$L=\begin{pmatrix}1&0&0&0&0&0&1&1\\1&1&0&0&0&0&0&1\\1&1&1&0&0&0&0&0\\0&1&1&1&0&0&0&0\\0&0&1&1&1&0&0&0\\0&0&0&1&1&1&0&0\\0&0&0&0&1&1&1&0\\0&0&0&0&0&1&1&1\end{pmatrix}$$

RoadRunneR 算法的其他信息详见文献[2]。

1.2　基于比特可分性的 MILP 模型

可分性是 Todo 在 2015 欧密会上提出的推广了的积分性质[3]，在 FSE 2016 上 Todo 和 Morri 又提出了比特可分性[5]，同时利用比特可分性找到了 SIMON32 的一个 14 轮积分区分器。在 2016 亚密会上，向泽军等人提出了基于比特可分性，利用 MILP 模型搜索积分区分器的方法[6]。克服了直接用比特可分性搜索区分器时，花费巨大时间和空间开销的问题。他们通过选择合适的目标函数，准确地表示可分性的传播，分析了 6 种具有比特置换扩散层的分组密码，但搜索扩散层为非比特置换的算法成为一个遗留的问题。孙玲等提出了解决这一问题的方法，通过引入一些中间变量，针对一个 n 比特输入的线性层，建立 2n 个线性不等式来表示穿过线性层的可分路径[7]。

利用基于比特可分性的 MILP 模型搜索积分区分器的总体步骤是：首先，给出初始比特可分性，即指定具体的活跃比特和非活跃比特；其次，建立表示穿过轮函数的可分路径的模型，包括穿过替换层以及扩散层。目前，对 S 盒可分路径的表示方法已经比较完善，而对线性层的可分路径表示方法还在进一步研究中[9]。最后，选择合适的目标函数，即搜索终止条件，对是否存在可用的积分区分器进行搜索。

2　预备知识

2.1　符号

令 $\mathbb{F}_2$ 表示二元有限域， $\mathbb{F}_2^n$ 表示在 $\mathbb{F}_2$ 上的 n 比特的序列。令 $\mathbb{Z}$ 和 $\mathbb{Z}^n$ 分别表示整数环和 n 维整数向量集合。对任意的 $a\in\mathbb{F}_2^n$，a[i] 表示 a 的第 i 个元素， $w(a)$ 表示汉明重量，其计算公式为 $a=(a_0,\cdots,a_{m-1})\in\mathbb{F}_2^{n_0}\times\cdots\times\mathbb{F}_2^{n_{m-1}}$，对任意的向量 $a=(a_0,\cdots,a_{m-1})\in\mathbb{F}_2^{n_0}\times\cdots\times\mathbb{F}_2^{n_{m-1}}$，向量 a 的汉明重量定义为 $W(a)=(w(a_0),\cdots,w(a_{m-1}))$．再令 $k=(k_0,k_1\cdots,k_{m-1})$ 和 $k^*=(k_0^*,k_1^*\cdots,k_{m-1}^*)$ 分别为 $\mathbb{Z}^m$ 上的两个向量。定义 $k\geq k^*$ 成立当且仅当向量中所有对应分量 $k_i\geq k_i^*$，其中 $i=0,1,\cdots,m-1$；否则 $k\not\geq k^*$。

比特乘积函数 $\pi_u(x)$ 和 $\pi_u(x)$　令 $\pi_u(x)$ 是一个从 $\mathbb{F}_2^n$ 到 $\mathbb{F}_2$ 的函数，对于任意的 $u\in\mathbb{F}_2^n$，使得 $x\in\mathbb{F}_2^n$ 是 π_u 的输入， $\pi_u(x)$ 定义式如下：

$$\pi_u(x):=\sum_{i=0}^{n-1}x[i]^{u[i]}$$

令 $\pi_u(x)$ 是一个 $\mathbb{F}_2^{n_0}\times\cdots\times\mathbb{F}_2^{n_{m-1}}$ 到 $\mathbb{F}_2$ 的函数，对于所有的 $u\in\mathbb{F}_2^{n_0}\times\cdots\times\mathbb{F}_2^{n_{m-1}}$，任意的 $u=(u_0,\cdots,u_{m-1})$，$x=(x_0,\cdots,x_{m-1})\in(\mathbb{F}_2^{n_0}\times\cdots\times\mathbb{F}_2^{n_{m-1}})$，定义式如下：

$$\pi_u(x):=\sum_{i=0}^{m-1}\pi_{u_i}(x_i)$$

2.2　可分性与可分路径

定义 1　可分性。令 $\mathbb{X}$ 为多重集合，其元素取值于 $(\mathbb{F}_2^n)^m$，k 是一个 m 维向量且每个分量取值于 0 到 n，当 $\mathbb{X}$ 满足可分性 $D_{k^{(0)},k^{(1)},\cdots k^{(q-1)}}^{n,m}$，需满足以下条件：对 $\mathbb{X}$ 中的任意一元素 x，$\pi_u(x)$ 的奇偶性始终为偶，当 $u\in\{(u_0,\cdots,u_{m-1})\in(\mathbb{F}_2^n)^m\mid W(u)\not\geq k^{(0)},\cdots,W(u)\not\geq k^{(q-1)}\}$.

定义 2　可分路径。令 f_r 表示一个分组密码的轮函数，假设分组密码的初始输入可分性为 $D_k^{n,m}$，再令经过 r 轮传播后的可分性为 $D_{\mathbb{K}_i}^{n,m}$，则可以得到以下的可分性传播链：

$$\{k\}\stackrel{\text{def}}{\Rightarrow}\mathbb{K}_0\xrightarrow{f_r}\mathbb{K}_1\xrightarrow{f_r}\mathbb{K}_2\xrightarrow{f_r}\cdots$$

对任意在 $\mathbb{K}_i(i\geq1)$ 中的向量 k_i^*，必然存在在 $\mathbb{K}_{i-1}$ 中的向量 k_{i-1}^* 与之对应，可以说 k_{i-1}^* 能通过可分性传播规则传播至 k_i^*，把它推广到更多维，记 $(k_0,k_1,\cdots,k_r)\in\mathbb{K}_0\times\mathbb{K}_1\times\cdots\times\mathbb{K}_r$，若对于所有的 $i\in\{1,2,\cdots r\}$，k_{i-1}^* 能传播至 k_i^*，本文称 $(k_0,k_1,\cdots,k_r)$ 为一个 r 轮的**可分路径**.

2.3　比特可分性传播规则与模型化

Todo 证明了传统可分性的传播规则[3]，并把这些传播规则总结在[4]中，分别是代替，拷贝，异或，分裂，合并五种操作的

传播规则。在比特可分性中，只用到其中拷贝和异或操作的传播规则。而对于 S 盒，即代替操作的比特可分性传播规则需要更加细致地研究，[8][10]研究了这一问题。在[6]中算法 2 给出了计算穿过 S 盒的比特可分性传播的通用算法。

规则 1　拷贝。 令 F 为一个拷贝函数，其输入 x 取值于 $\mathbb{F}_2$，输出可表示为 $(y_0,y_1)=(x,x)$．令 $\mathbb{X}$ 和 $\mathbb{Y}$ 分别为对应的输入和输出集合，假设 $\mathbb{X}$ 满足可分性 D_k^1,则 $\mathbb{Y}$ 满足可分性 $D_{\mathbb{K}}^{1\times1}$．传播过程只有两种可能情况。

$$\begin{cases} k=0\rightarrow\mathbb{K}=\{(0,0)\} \\ k=1\rightarrow\mathbb{K}=\{(0,1),(1,0)\} \end{cases}$$

规则 2　异或压缩。 令 F 为一个异或压缩函数，其输入 (x_0,x_1) 取值于 $\mathbb{F}_2\times\mathbb{F}_2$，输出可表示为 $y=x_0\oplus x_1$,令 $\mathbb{X}$ 和 $\mathbb{Y}$ 分别为对应的输入和输出集合，假设 $\mathbb{X}$ 满足可分性 $D_k^{1\times1}$,则 $\mathbb{Y}$ 满足可分性 $D_{\mathbb{K}}^1$.传播过程只有四种可能情况。

$$\begin{cases} k=(0,0)\rightarrow\mathbb{K}=\{(0)\} \\ k=(0,1)\rightarrow\mathbb{K}=\{(1)\} \\ k=(1,0)\rightarrow\mathbb{K}=\{(1)\} \\ k=(1,1)\rightarrow\mathbb{K}=\phi \end{cases}$$

规则 3　与压缩。令 F 为一个与压缩函数，其输入 (x_0,x_1) 取值于 $\mathbb{F}_2\times\mathbb{F}_2$，输出可表示为 $y=x_0\wedge x_1$,令 $\mathbb{X}$ 和 $\mathbb{Y}$ 分别为对应的输入和输出集合，假设 $\mathbb{X}$ 满足可分性 $D_k^{1\times1}$,则 $\mathbb{Y}$ 满足可分性 $D_{\mathbb{K}}^1$.传播过程只有四种可能情况。

$$\begin{cases} k=(0,0)\rightarrow\mathbb{K}=\{(0)\} \\ k=(0,1)\rightarrow\mathbb{K}=\{(1)\} \\ k=(1,0)\rightarrow\mathbb{K}=\{(1)\} \\ k=(1,1)\rightarrow\mathbb{K}=\{(1)\} \end{cases}$$

下面简要地说明用线性不等式组对拷贝，与，异或操作的可分性传播的建模过程。

模型 1　拷贝。 记 $(a)\xrightarrow{copy}(b_0,b_1)$ 是拷贝函数的一条可分路径，则下列不等式组可准确表示拷贝操作的可分性传播。

$$\begin{cases} a-b_0-b_1=0 \\ a,b_0,b_1\text{是二进制数} \end{cases}$$

模型 2　与。 记 $(a_0,a_1)\xrightarrow{And}(b)$ 是比特与函数的一条可分路径，则下列不等式组可准确表示比特与操作的可分性传播。

$$\begin{cases} b-a_0\geq0 \\ b-a_1\geq0 \\ b-a_0-a_1\geq0 \\ a_0,a_1,\text{b是二进制数} \end{cases}$$

模型 3　异或。记 $(a_0,a_1)\xrightarrow{Xor}(b)$ 是比特异或函数的一条可分路径,则下列不等式组可准确表示比特异或操作的可分性传播。

$$\begin{cases} a_0+a_1-b=0 \\ a_0,\text{a}_1,\text{b是二进制数} \end{cases}$$

模型化 S 盒　对于 S 盒的传播，向泽军等给出计算穿过 S 盒的比特可分性传播过程[6]。然后通过 Sage 软件中的 Inequality_generator()函数，该函数返回一组线性不等式，这组线性不等式还可以通过缩减算法来缩减不等式个数[6]。

2.4　目标函数

若一个集合 $\mathbb{X}$ 满足可分性 $D_{\mathbb{K}}^{1,n}$，$\mathbb{X}$ 不存在积分性质当且仅当 $\mathbb{K}$ 包含所有的 n 个单位向量。记经过 i 轮加密后的输出可分性为 $D_{\mathbb{K}_i}^{n,m}$,若 $\mathbb{K}_{r+1}$ 是第一次出现包含了所有的 n 个单位向量,可分性的传播终止，这样得到了一个 r 轮区分器。判断是否存在 r-1 轮区分器，只需检测 $\mathbb{K}_r$ 是否包含全部的单位向量，因此设定目标函数：

$$\text{Obj:Min}\{a_0{}^r+a_1{}^r+\cdots+a_{n-1}{}^r\}$$

其中：$a_0{}^r,a_1{}^r,\cdots,a_{n-1}{}^r$ 是一个任意 r 轮可分路径的最后一个向量。

3　PRIDE 的 MILP 模型

在可分性的传播过程中，与常数异或不改变可分性，即轮密钥加不会影响可分性的传播，因此只考虑 S 盒和线性扩散层对可分性传播的影响。

3.1　S 层的线性不等式表示

PRIDE 的 S 层由 16 个相同的 S 盒并置而成，首先研究穿过 S 盒的可分路径。令 S 盒的输入为 $x=(x_3,x_2,x_1,x_0)$，对应输出为 $y=(y_3,y_2,y_1,y_0)$,则 S 盒的代数规范式（ANF）表示如下：

$$\begin{cases} y_0=x_2\oplus x_0x_1\oplus x_1x_2\oplus x_0x_2x_3 \\ y_1=x_3\oplus x_0x_1\oplus x_1x_2\oplus x_0x_2x_3\oplus x_1x_2x_3 \\ y_2=x_0\oplus x_1x_2 \\ y_3=x_1\oplus x_2x_3 \end{cases}$$

应用算法 2 计算穿过 S 盒的可分路径，可得到总共 44 条可分路径，如表 3 所示。

表 3　PRIDE 的 S 盒可分路径

Table 3　S box divisible path of PRIDE

输入 $D_k^{1,4}$	输出 $D_{\mathbb{K}}^{1,4}$
(0,0,0,0)	(0,0,0,0)
(0,0,0,1)	(0,0,0,1)　(0,0,1,0)　(0,1,0,0)
(0,0,1,0)	(0,0,0,1)　(0,0,1,0)　(0,1,0,0)　(1,0,0,0)
(0,0,1,1)	(0,0,0,1)　(0,0,1,0)　(1,1,0,0)
(0,1,0,0)	(0,0,0,1)　(0,0,1,0)　(0,1,0,0)　(1,0,0,0)
(0,1,0,1)	(0,0,0,1)　(0,0,1,0)　(1,1,0,0)
(0,1,1,0)	(0,0,0,1)　(0,0,1,0)　(0,1,0,0)
(0,1,1,1)	(0,0,1,1)　(1,1,0,1)
(1,0,0,0)	(0,0,0,1)　(0,0,1,0)　(1,0,0,0)
(1,0,0,1)	(0,0,0,1)　(0,0,1,0)　(1,1,0,0)
(1,0,1,0)	(0,0,1,0)　(1,0,0,1)　(1,1,0,0)
(1,0,1,1)	(1,0,0,1)　(1,1,1,0)
(1,1,0,0)	(0,0,0,1)　(0,0,1,0)　(1,0,0,0)
(1,1,0,1)	(0,0,0,1)　(0,0,1,0)　(1,1,0,0)
(1,1,1,0)	(0,0,1,0)　(1,0,0,1)　(1,1,0,0)
(1,1,1,1)	(1,1,1,1)

利用 Sage 软件中的 Inequality_generator()函数，共返回 110 个线性不等式表示这 44 条可分路径，再通过不等式缩减算法把此组不等式个数缩减为 11 个，记 $(a_0,a_1,a_2,a_3)\xrightarrow{S}(b_0,b_1,b_2,b_3)$ 表示一条穿过 S 盒的可分路径，则它的线性不等式 $\mathcal{L}_S$ 表示如下：

$$\mathcal{L}_S:\begin{cases} a_0+a_1+a_2+a_3-b_0-b_1-b_2-b_3\geq 0\\ -a_0+3b_0-2b_1-b_2-b_3+2\geq 0\\ -a_1-a_2-2a_3+b_0+2b_1+3b_2-3b_3\geq 0\\ 2a_0+a_1+a_2-3b_0+b_1-2b_2-2b_3+2\geq 0\\ -a_1-b_0-b_2+b_3+2\geq 0\\ -2a_0-2a_2-a_3+2b_0-b_1+b_2+3\geq 0\\ a_2+a_3-b_0-b_1-b_3+1\geq 0\\ -a_0-a_2-b_0+b_1+b_3+2\geq 0\\ a_0+2a_3-b_0-b_1-b_2-b_3+1\geq 0\\ -a_1-a_3-b_0+b_1+2\geq 0\\ -a_0+b_0+b_2+b_3\geq 0\\ a_i,b_i\text{为二进制数} \end{cases}$$

对整个 S 层的可分路径线性不等式表示，可直接把 16 个 4 维输入向量并置，输出同样为 16 个 4 维向量。综上，记 $(x_0,x_1,\cdots,x_{63})\xrightarrow{S}(y_0,y_1,\cdots,y_{63})$ 为穿过 S 层的一条可分路径。若给出一个 S 层输入集合满足可分性 $D_k^{1,64}$，则穿过 S 层后输出满足可分性 $D_{\mathbb{K}}^{1,64}$.

3.2 线性层的线性不等式表示

利用 MILP 模型为搜索更复杂线性层的算法成为一个遗留的问题，可以引入一些中间变量解决这个问题：任何线性矩阵层，都可以分割为拷贝和异或这两个操作[7]。

PRIDE 的 64×64 矩阵层由四个 16×16 小矩阵呈对角线排列构成，分别为 L_0,L_1,L_2,L_3 .现以 L_0 为例，得出关于 L_0 的线性不等式组，其余三个矩阵过程类似。

假设穿过矩 L_0 的输入集合满足可分性 $D_x^{1,16}$，其中 $x=(x_0,x_1,\cdots,x_{15})$.从 L_0 的第一列看出，x_0 被拷贝了 5 次，这些被拷贝的值又在不同的行和其他被拷贝的进行异或，剩余的 $x_i(1\leq i\leq 15)$ 和 x_0 的操作类似。

在 L_0 中共有 48 个非零元素，通过引入中间变量 $t_0\sim t_{47}$，L_0 转换为如下形式：

$$\begin{pmatrix}
0&0&0&0&t_{12}&0&0&0&t_{24}&0&0&0&t_{36}&0&0&0\\
0&0&0&0&0&t_{15}&0&0&0&t_{27}&0&0&0&t_{39}&0&0\\
0&0&0&0&0&0&t_{18}&0&0&0&t_{30}&0&0&0&t_{42}&0\\
0&0&0&0&0&0&0&t_{21}&0&0&0&t_{33}&0&0&0&t_{45}\\
t_0&0&0&0&0&0&0&0&t_{25}&0&0&0&t_{37}&0&0&0\\
0&t_3&0&0&0&0&0&0&0&t_{28}&0&0&0&t_{40}&0&0\\
0&0&t_6&0&0&0&0&0&0&0&t_{31}&0&0&0&t_{43}&0\\
0&0&0&t_9&0&0&0&0&0&0&0&t_{34}&0&0&0&t_{46}\\
t_1&0&0&0&t_{13}&0&0&0&0&0&0&0&t_{38}&0&0&0\\
0&t_4&0&0&0&t_{16}&0&0&0&0&0&0&0&t_{41}&0&0\\
0&0&t_7&0&0&0&t_{19}&0&0&0&0&0&0&0&t_{44}&0\\
0&0&0&t_{10}&0&0&0&t_{22}&0&0&0&0&0&0&0&t_{47}\\
t_2&0&0&0&t_{14}&0&0&0&t_{26}&0&0&0&0&0&0&0\\
0&t_5&0&0&0&t_{17}&0&0&0&t_{29}&0&0&0&0&0&0\\
0&0&t_8&0&0&0&t_{20}&0&0&0&t_{32}&0&0&0&0&0\\
0&0&0&t_{11}&0&0&0&t_{23}&0&0&0&t_{35}&0&0&0&0
\end{pmatrix}$$

为了描述 L_0 所有的拷贝操作，生成以下 16 个线性不等式：

$$(1)\begin{cases} x_0-t_0-t_1-t_2=0\\ x_1-t_3-t_4-t_5=0\\ x_2-t_6-t_7-t_8=0\\ x_3-t_9-t_{10}-t_{11}=0\\ x_4-t_{12}-t_{13}-t_{14}=0\\ x_5-t_{15}-t_{16}-t_{17}=0\\ x_6-t_{18}-t_{19}-t_{20}=0\\ x_7-t_{21}-t_{22}-t_{23}=0\\ x_8-t_{24}-t_{25}-t_{26}=0\\ x_9-t_{27}-t_{28}-t_{29}=0\\ x_{10}-t_{30}-t_{31}-t_{32}=0\\ x_{11}-t_{33}-t_{34}-t_{35}=0\\ x_{12}-t_{36}-t_{37}-t_{38}=0\\ x_{13}-t_{39}-t_{40}-t_{41}=0\\ x_{14}-t_{42}-t_{43}-t_{44}=0\\ x_{15}-t_{45}-t_{46}-t_{47}=0\\ x_i,t_i\text{为二进制数} \end{cases}$$

另一方面，x_i 的拷贝比特需要和相关的输出比特异或。记 $(x_0,x_1,\cdots,x_{15})\xrightarrow{L_0}(y_0,y_1,\cdots,y_{15})$ 是穿过 L_0 的一条可分路径，可以看出，在同一行的变量和需要异或的变量是相同的。为了描述 L_0 所有的异或操作，生成以下 16 个线性不等式：

$$(2)\begin{cases} t_{12}+t_{24}+t_{36}-y_0=0\\ t_{15}+t_{27}+t_{39}-y_1=0\\ t_{18}+t_{30}+t_{42}-y_2=0\\ t_{21}+t_{33}+t_{45}-y_3=0\\ t_0+t_{25}+t_{37}-y_4=0\\ t_3+t_{28}+t_{40}-y_5=0\\ t_6+t_{31}+t_{43}-y_6=0\\ t_9+t_{34}+t_{46}-y_7=0\\ t_1+t_{13}+t_{38}-y_8=0\\ t_4+t_{16}+t_{41}-y_9=0\\ t_7+t_{19}+t_{44}-y_{10}=0\\ t_{10}+t_{22}+t_{47}-y_{11}=0\\ t_2+t_{14}+t_{26}-y_{12}=0\\ t_5+t_{17}+t_{29}-y_{13}=0\\ t_8+t_{20}+t_{32}-y_{14}=0\\ t_{11}+t_{23}+t_{35}-y_{15}=0\\ y_i,t_i\text{为二进制数} \end{cases}$$

为了得到穿过 L_0 的可分路径，仅需要把以上两组线性不等式组(1)(2)联合起来成为一个线性不等式组 $\mathcal{L}_0$ 。

类似地，对穿过 $(x_{16},x_{17},\cdots,x_{31})\xrightarrow{L_1}(y_{16},y_{17},\cdots,y_{31})$ 的可分路径分别表示为 $(x_{16},x_{17},\cdots,x_{31})\xrightarrow{L_1}(y_{16},y_{17},\cdots,y_{31})$，$(x_{32},x_{33},\cdots,x_{47})\xrightarrow{L_2}(y_{32},y_{33},\cdots,y_{47})$，$(x_{48},x_{49},\cdots,x_{63})\xrightarrow{L_3}(y_{48},y_{49},\cdots,y_{63})$ 中间变量分别表示为 $t_{48}\sim t_{95}$，$t_{96}\sim t_{143}$，$t_{144}\sim t_{191}$.按照以上的传播规则，分别生成对应的线性不等式组 $\mathcal{L}_1,\mathcal{L}_2,\mathcal{L}_3$ 。

综上，记 $(x_0,x_1,\cdots,x_{63})\xrightarrow{L}(y_0,y_1,\cdots,y_{63})$ 为穿过线性矩阵层的一条可分路径。若给出一个线性层输入集合满足可分性 $D_k^{1,64}$，则穿过线性层后输出满足可分性 $D_{\mathbb{K}}^{1,64}$.

得到了穿过 S 层和 L 层的可分路径线性不等式组表示，结合起来便得到轮函数的可分路径，重复轮函数 r 次，即得到 r 轮的可分路径。

4　RoadRunneR 的 MILP 模型

RoadRunner 算法总体采用 Feistel 结构，包含拷贝和异或操作，根据 1 中提到的方法对这两种操作建模。轮函数比较复杂，但总体来看，依然是 S 层和线性扩散层构成，类似于 2 中的步骤对其建模。

RoadRunner 算法的第一轮和最后一轮采用了白化密钥，且每一轮加入了轮常量异或运算，但由于与常数异或不改变可分性，故将这些运算忽略。

4.1　S 层的线性不等式表示

RoadRunner 的 S 层由 8 个相同的 S 盒并置而成，令 S 盒的输入为 $x=(x_3,x_2,x_1,x_0)$，对应输出为 $y=(y_3,y_2,y_1,y_0)$,则 S 盒的代数规范式（ANF）表示如下：

$$\begin{cases} y_0 = x_2 \oplus x_0x_1 \\ y_1 = x_1 \oplus x_0x_1 \oplus x_0x_2 \oplus x_0x_3 \oplus x_0x_1x_2 \\ y_2 = x_1 \oplus x_2 \oplus x_3 \oplus x_1x_2 \\ y_3 = x_0 \oplus x_2x_3 \oplus x_0x_1x_3 \end{cases}$$

应用算法 2，可得到总共 43 条可分路径，如表 4 所示。

表 4　RoadRunneR 的 S 盒可分路径

Table 4　S box divisible path of RoadRunneR

输入 $D_k^{1,4}$	输出 $D_{\mathbb{K}}^{1,4}$
(0,0,0,0)	(0,0,0,0)
(0,0,0,1)	(0,0,0,1)　(0,0,1,0)　(1,0,0,0)
(0,0,1,0)	(0,0,0,1)　(0,0,1,0)　(0,1,0,0)　(1,0,0,0)
(0,0,1,1)	(0,0,0,1)　(0,0,1,0)　(1,0,0,0)
(0,1,0,0)	(0,0,0,1)　(0,0,1,0)　(0,1,0,0)　(1,0,0,0)
(0,1,0,1)	(0,0,1,0)　(1,0,0,1)　(1,1,0,0)
(0,1,1,0)	(0,0,1,0)　(0,1,0,0)
(0,1,1,1)	(0,0,1,0)　(1,1,0,0)
(1,0,0,0)	(0,0,1,0)　(0,1,0,0)　(1,0,0,0)
(1,0,0,1)	(0,0,1,0)　(0,1,0,1)　(1,0,0,0)
(1,0,1,0)	(0,0,1,1)　(0,1,0,1)　(0,1,1,0)　(1,0,0,0)
(1,0,1,1)	(0,0,1,1)　(0,1,0,1)　(1,0,0,0)
(1,1,0,0)	(0,0,1,1)　(0,1,0,1)　(1,0,0,0)
(1,1,0,1)	(0,0,1,1)　(1,1,0,1)
(1,1,1,0)	(0,1,1,1)　(1,0,1,0)
(1,1,1,1)	(1,1,1,1)

利用 Sage 软件中的 Inequality_generator()函数，共返回 132 个线性不等式表示这 43 条可分路径，再通过不等式缩减算法把此组不等式个数缩减为 10 个，记 $(a_0,a_1,a_2,a_3)\xrightarrow{S}(b_0,b_1,b_2,b_3)$ 表示一条穿过 S 盒的可分路径，则它的线性不等式 $\mathcal{L}_S$ 表示如下：

$$\mathcal{L}_S:\begin{cases} a_0+a_1+a_2+a_3-b_0-b_1-b_2-b_3 \ge 0 \\ -4a_0-3a_1-a_2-2a_3+2b_0-b_1+b_2+3b_3+5 \ge 0 \\ -2a_0-4a_1-3a_2-a_3+b_0+2b_1+3b_2-b_3+5 \ge 0 \\ 3a_1-2b_0-b_1-b_2-b_3+2 \ge 0 \\ 2a_0-b_1-b_2-b_3+1 \ge 0 \\ -2a_0-a_1-a_2+3b_0+2b_1+2b_2+b_3 \ge 0 \\ -a_0-2a_1-2a_3-b_0+b_1+b_3+4 \ge 0 \\ a_3-b_0-b_1-b_2+1 \ge 0 \\ -a_3+b_0+b_2+b_3 \ge 0 \\ a_1+a_2-b_0-b_2-b_3+1 \ge 0 \\ a_i,b_i\text{为二进制数} \end{cases}$$

综上，记 $(x_0,x_1,\cdots,x_{31})\xrightarrow{S}(y_0,y_1,\cdots,y_{31})$ 为穿过 S 层的一条可分路径。若给出一个 S 层输入集合满足可分性 $D_k^{1,32}$，则穿过 S 层后输出满足可分性 $D_{\mathbb{K}}^{1,32}$.

4.2　线性层的线性不等式表示

RoadRunneR 的矩阵层由四个相同的 8×8 矩阵 L 并置构成，以下得出关于 L 的线性不等式组。

假设穿过矩阵 L 的输入集合满足可分性 $D_x^{1,8}$，其中 $x=(x_0,x_1,\cdots,x_7)$. 在 L 中共有 24 个非零元素，通过引入中间变量 $t_0\sim t_{23}$, L 转换为如下形式：

$$\begin{pmatrix} t_0 & 0 & 0 & 0 & 0 & 0 & t_{18} & t_{21} \\ t_1 & t_3 & 0 & 0 & 0 & 0 & 0 & t_{22} \\ t_2 & t_4 & t_6 & 0 & 0 & 0 & 0 & 0 \\ 0 & t_5 & t_7 & t_9 & 0 & 0 & 0 & 0 \\ 0 & 0 & t_8 & t_{10} & t_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & t_{11} & t_{13} & t_{15} & 0 & 0 \\ 0 & 0 & 0 & 0 & t_{14} & t_{16} & t_{19} & 0 \\ 0 & 0 & 0 & 0 & 0 & t_{17} & t_{20} & t_{23} \end{pmatrix}$$

为了描述 L 中所有的拷贝操作，生成以下 8 个线性不等式：

$$(1)\begin{cases} x_0-t_0-t_1-t_2 = 0 \\ x_1-t_3-t_4-t_5 = 0 \\ x_2-t_6-t_7-t_8 = 0 \\ x_3-t_9-t_{10}-t_{11} = 0 \\ x_4-t_{12}-t_{13}-t_{14} = 0 \\ x_5-t_{15}-t_{16}-t_{17} = 0 \\ x_6-t_{18}-t_{19}-t_{20} = 0 \\ x_7-t_{21}-t_{22}-t_{23} = 0 \\ x_i, t_i\text{为二进制数} \end{cases}$$

记 $(x_0,x_1,\cdots,x_7)\xrightarrow{L}(y_0,y_1,\cdots,y_7)$ 是穿过 L 的一条可分路径，为了描述 L 所有的异或操作，生成以下 8 个线性不等式：

$$(2)\begin{cases} t_0+t_{18}+t_{21}-y_0 = 0 \\ t_1+t_3+t_{22}-y_1 = 0 \\ t_2+t_4+t_6-y_2 = 0 \\ t_5+t_7+t_9-y_3 = 0 \\ t_8+t_{10}+t_{12}-y_4 = 0 \\ t_{11}+t_{13}+t_{15}-y_5 = 0 \\ t_{14}+t_{16}+t_{19}-y_6 = 0 \\ t_{17}+t_{20}+t_{23}-y_7 = 0 \\ y_i, t_i \text{为二进制数} \end{cases}$$

综上，记 $(x_0,x_1,\cdots,x_{31})\xrightarrow{L}(y_0,y_1,\cdots,y_{31})$ 为穿过线性矩阵层的一条可分路径。若给出一个线性层输入集合满足可分性 $D_k^{1,32}$，则穿过线性层后输出满足可分性 $D_{\mathbb{K}}^{1,32}$.

得到了穿过 S 层和 L 层的可分路径线性不等式组表示，结合起来便得到 SLK 函数的可分路径。结合三个 SLK 函数和一个 S 层，可得到一轮函数的可分路径。重复轮函数 r 次，即得到 r 轮的可分路径。

5 积分区分器搜索

在前面分别得到针对 PRIDE 和 RoadRunner 算法完整的 MILP 模型，$\mathcal{L}$ 为约束条件，Obj 为目标函数。根据搜索算法[6]，应用 Gurobi 优化器求解该模型，可搜索是否存在 r 轮积分区分器。本节中的实验数据均来自以下平台：Intel(R) Celeron(R)CPU 1007U @1.50 GHz,4 GB RAM,64 bit Windows 7。采用 Python 编程实现搜索区分器得到如表 5 所示。

表 5 搜索的积分区分器

Table 5 Integral differentiator for Search

密码算法	分组长度	区分器轮数	活跃比特	平衡比特
PRIDE	64	4	16	64
		6	40	16
		8	60	32
		9	63	1
RoadRunneR	64	4	32	32
		5	50	32

5.1 PRIDE 的 8 轮积分区分器

若选择输入的初始可分性为 $D^{1,64}_{0fffffffffffffff}$,即选择输入的右边 60 bit 为活跃比特，其余 4 bit 为常数，经过 8 轮之后得到一个有效的区分器，区分器输出的右边 32 比特为平衡比特。PRIDE 的 8 轮区分器可表示如下：

输入:

(ccccaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaa)

输出:

(????????????????????????????????bbbbbbbbbbbbbbbbbbbb bbbbbbbbbbbb)

5.2 PRIDE 的 9 轮积分区分器

若选择输入的初始可分性为 $D^{1,64}_{7fffffffffffffff}$,即选择输入的右边 63 比特为活跃比特，经过 9 轮之后得到一个有效的区分器，区分器输出的最右边比特为平衡比特。PRIDE 的 9 轮区分器可表示如下：

输入:

(ccccaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaa)

输出:

(?? ????????b)

5.3 RoadRunneR 的 5 轮积分区分器

若选择输入的初始可分性为 $D^{1,64}_{0003ffff,ffffffff}$,即选择输入的右边 32 bit 均为活跃比特，左边 18 比特为活跃比特，其余 14 bit 为常数，经过 5 轮之后得到一个有效的区分器，区分器输出的右边 32 bit 为平衡比特。RoadRunner 的 5 轮区分器可表示如下：

输入：

(ccccccccccccccaaaaaaaaaaaaaaaaaa,aaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaa)

输出:

(????????????????????????????????,bbbbbbbbbbbbbbbbbbbbb bbbbbbbbbbb)

同时搜索到有一个 RoadRunneR 4 轮区分器，与算法设计者在原文中提出的 4 轮区分器一致，可证明利用该方法搜索积分区分器结果的正确性。

6 结束语

之前有人对 PRIDE 算法做过了差分分析和线性分析[13][14]，但 PRIDE 算法的积分分析还没有人研究。RoadRunneR 算法在设计者的原文章中提出有一个 4 轮区分器。本文利用基于比特可分性的 MILP 模型搜索 PRIDE 算法的积分区分器，得到了 9 轮区分器，该区分器是目前对 PRIDE 算法最长积分区分器。应用此方法搜索 RoadRunneR 算法的积分区分器，得到了 5 轮区分器，对比设计者的区分器提升了一轮。可以看出无论 Feistel 还是 SPN 型的分组密码，对比传统寻找区分器的方法，应用该技术均可以搜索出较为理想的积分区分器。

利用以上方法得到了目前已知的最长积分区分器，可对这两种密码算法进行目前已知最多轮的积分攻击。例如对于 PRIDE 的 8 轮区分器，向后扩展两轮可进行 10 轮攻击，根据相关密钥恢复的方法与技术，恢复相应的密钥。

但是增加积分区分器长度的同时会增加选择明文数量，这样在整个攻击过程中的数据复杂度并不会有显著的降低。例如搜索 PRIDE 算法得到的 9 轮区分器，需要选择明文数为 2^{63} 个。因此对于同样轮数长度的区分器，降低选择明文数量是寻找区分器的难点。从当前研究来看，利用比特可分性搜索到的积分区分器在大多数情况是最优的区分器，但并不能说一定是最优的区分器，这关系到可分性传播过程的模型化表示方法的准确性，也是下一步研究的主要方向。



随着对比特可分性和自动化搜索区分器的研究进行，结合数学模型和优化软件提高了搜索的效率，大大改进了寻找积分区分器的方法，因而对积分分析也有了很大的推进，而且基于比特可分性搜索区分器还有很大的潜力可挖，本文也是对这一方法的研究和应用。

参考文献：

[1] Albrecht M, Driessen B, Kavun E B, *et al.* Block ciphers focus on the linear layer (feat. PRIDE) [C/OL]. IACR Cryptology ePrint Archive, (2014) https://eprint. iacr. org/2014/453. pdf.

[2] Baysal A, Sahin S. RoadRunneR: A small and fast bitslice block cipher for low cost 8-bit processors [C/OL]. LightSec Cryptology ePrint Archive. (2015) . https://eprint. iacr. org/2015/906. pdf

[3] Todo Y. Structural evaluation by generalized integral property [C]// Proc of EUROCRYPT, LNCS, vol. 9056. 2015. Berlin: Springer, 2015: 287–314.

[4] Todo Y. Integral cryptanalysis on full MISTY1 [C]// Proc of CRYPTO Part I. LNCS vol. 9215. Berlin: Springer, 2015: 413–432.

[5] Todo Y, Morii M. Bit-based division property and application to simon family [C]// Pre-Proceedings of FSE, 2016. Berlin: Springer, 2015: 357–377.

[6] Xiang Zejun, Zhang Wentao, Bao Zhenzhen *et al.* Applying milp method to search integral distinguishers based on division property for 6 lightweight block ciphers [C]// Advances in Cryptology Berlin: Springer, 2016: 648-678.

[7] Sun Ling, Wang Wei, Wang Meiqin. Milp-aided bit-based division property for primitives with non-bit-permutation linear layers [C/OL]. Cryptology ePrint Archive. (2016) http://eprint. iacr. org/2016/811.

[8] Sun Ling, Wang Meiqin. Towards a further understanding of bit-based division property [J]. Science China Information Sciences, 2017, 60 (12): 128101.

[9] Zhang Wenying, Rijmen V. Division cryptanalysis of block ciphers with a binary diffusion layer [C/OL]. IACR Cryptology ePrint Archive. (2017) https://eprint. iacr. org/2017/188.

[10] Boura C, Canteaut A. Another view of the division property [C]// Advances in Cryptology-CRYPTO. Berlin: Springer, 2016: 654-682.

[11] Daemen J, Knudsen L, Rijmen V. The block cipher square [C]// Proc of the 4th International Workshop on Fast Software Encryption. Berlin: Springer, 1997: 149-165.

[12] Knudsen L, Wagner D. Integral cryptanalysis [C]// Proc of the 9th Int Workshop on Fast Software Encryption, 2002 [C]. Berlin: Springer, 2002: 112-127.

[13] Dai Yibin, Chen Shaozhen. Cryptanalysis of full PRIDE block cipher [J]. Science China Information Sciences. 2017, 60 (5): 052108.

[14] 伊文坛，陈少真，田亚. 减缩轮 PRIDE 算法线性分析 [J]. 电子学报, 2017, 45 (2): 468-476. (Yi Wentan, Chen Shaozhen, Tian Ya. Linear cryptanalysis of reduced-round PRIDE block cipher [J]. Acta Electronica Sinica, 2017, 45 (2): 468-476.)

[15] 于晓丽，吴文玲，李艳俊. 低轮 MIBS 分组密码的积分分析 [J]. 计算机研究与发展, 2013, 50 (10): 2117-2125. (Yu Xiaoli, Wu Wenling, Li Yanjun. Integral attack of reduced-round MIBS block cipher [J]. Journal of Computer Research & Development, 2013, 50 (10): 2117-2125.)